(12) UK Patent Application (19) GB (11) 2 305 827 (13) A

(43) Date of A Publication 16.04.1997

(21) Application No 9519995.6

(22) Date of Filing 30.09.1995

(71) Applicant(s)
Motorola Limited

(Incorporated in the United Kingdom)

European Intellectual Property Operation, Jays Close, Viables Industrial Estate, BASINGSTOKE, Hampshire, RG22 4PD, United Kingdom

(72) Inventor(s)
Paul Adrian Golding
Irwin Russell Bennett

(74) Agent and/or Address for Service
Peter D Hudson
Motorola Limited, European Intellectual Property Operation, Midpoint, Alencon Link, BASINGSTOKE, Hampshire, RG21 7PL, United Kingdom

(51) INT CL^6
H04L 1/00 , H03M 13/00

(52) UK CL (Edition O)
H4P PRV

(56) Documents Cited
None

(58) Field of Search
UK CL (Edition N) **H4P PRV**
INT CL^6 **H03M 13/00 , H04L 1/00**
Online:WPI,CLAIMS,JAPIO,USPATFULL,INSPEC

(54) Viterbi trellis decoder reduces number of metric difference calculations

(57) The Viterbi state trellis (120) was Window Erro: Detection and effects two passes of a frame through the trellis (120). During a first pass, metric-difference calculations are suspended, but path metric calculations are indexed. At an identified branch, trellis data words (138-144) are recorded and associated with the branch. Upon completion of the first pass, a desired survivor path is determined and "traced-back" to its corresponding state in the identified branch using indexed path metrics, for example. The path metric corresponding to this state is nominally reloaded and, starting from that state, decoding of the frame occurs for a second pass while metric-difference (L-value) calculations are performed on the desired survivor path only, and only a single lowest L-value for the desired survivor path is recorded. This is intended to reduce memory space for the calculations and reduce power requirements. It may be used in GSM cellular communications using forward error correction.

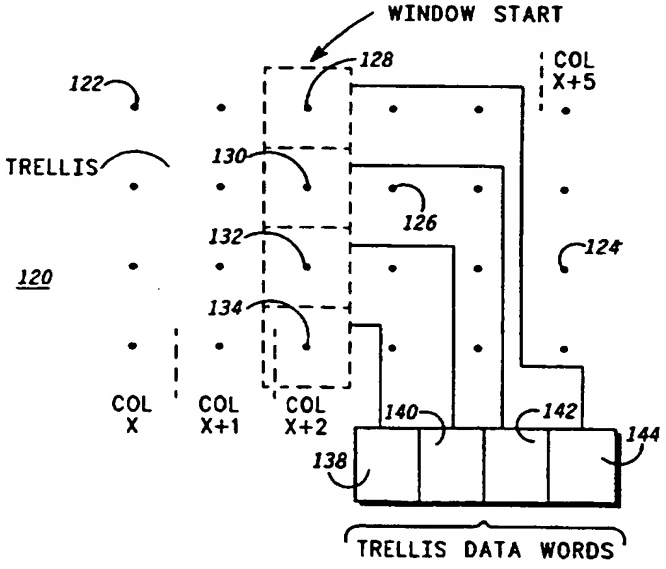
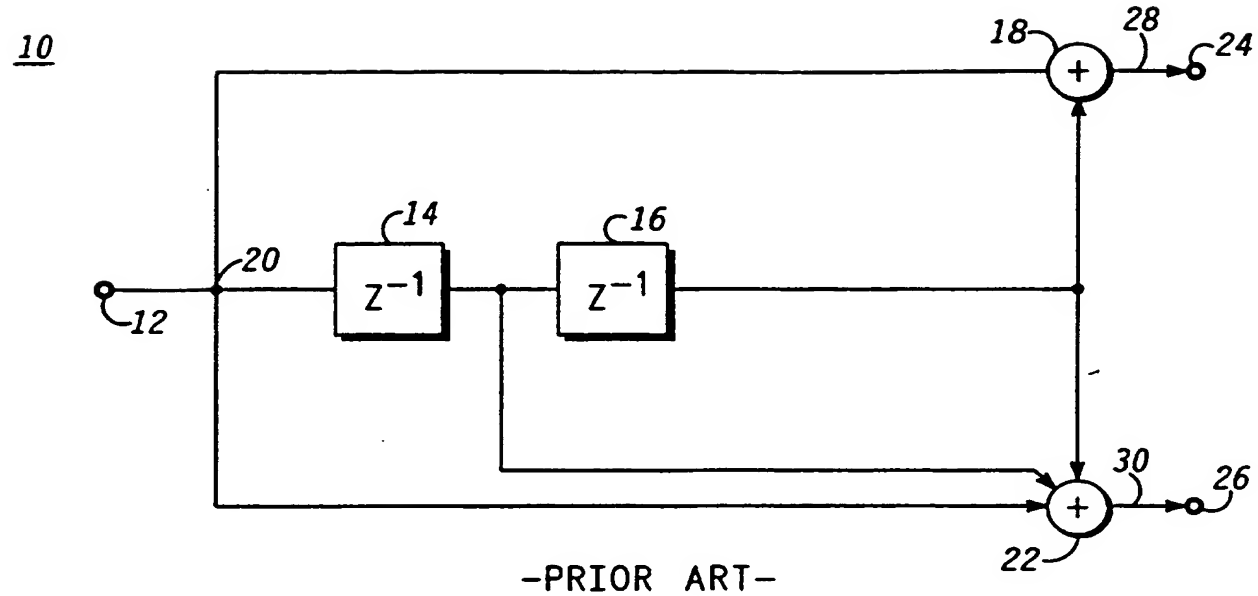


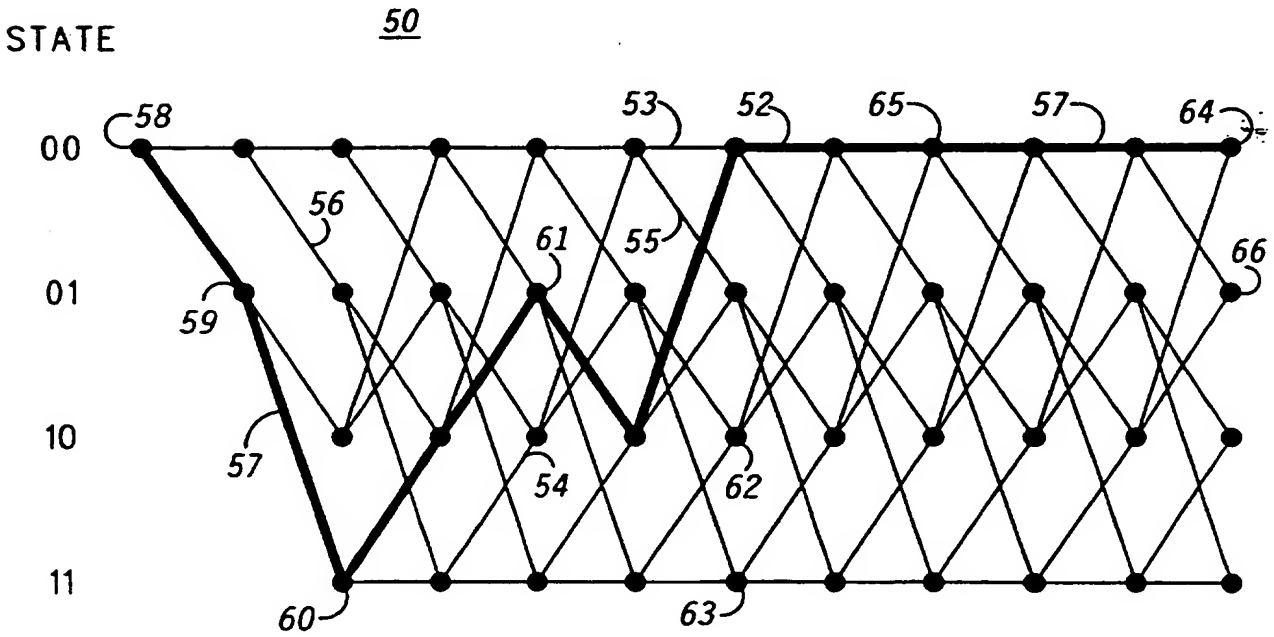
FIG. 7

GB 2 305 827 A



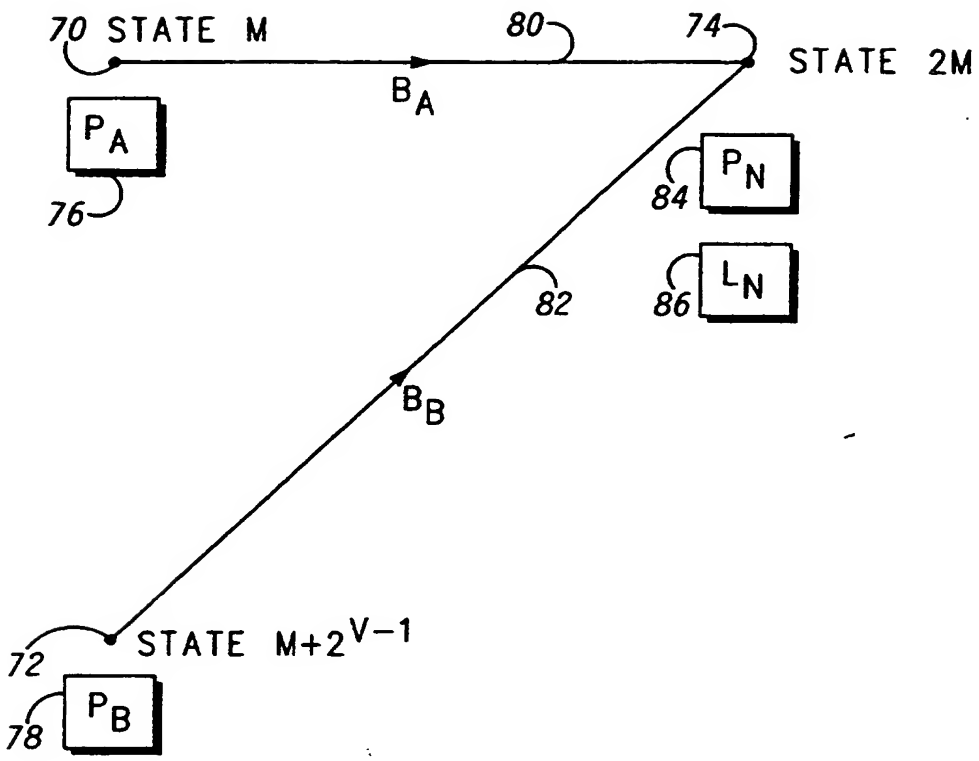
-PRIOR ART-

FIG. 1



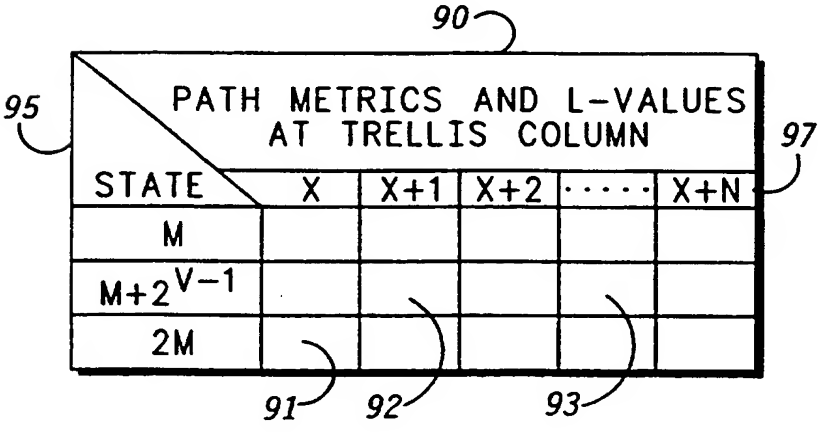
-PRIOR ART-

FIG. 2



-PRIOR ART-

FIG. 3



-PRIOR ART-

FIG. 4

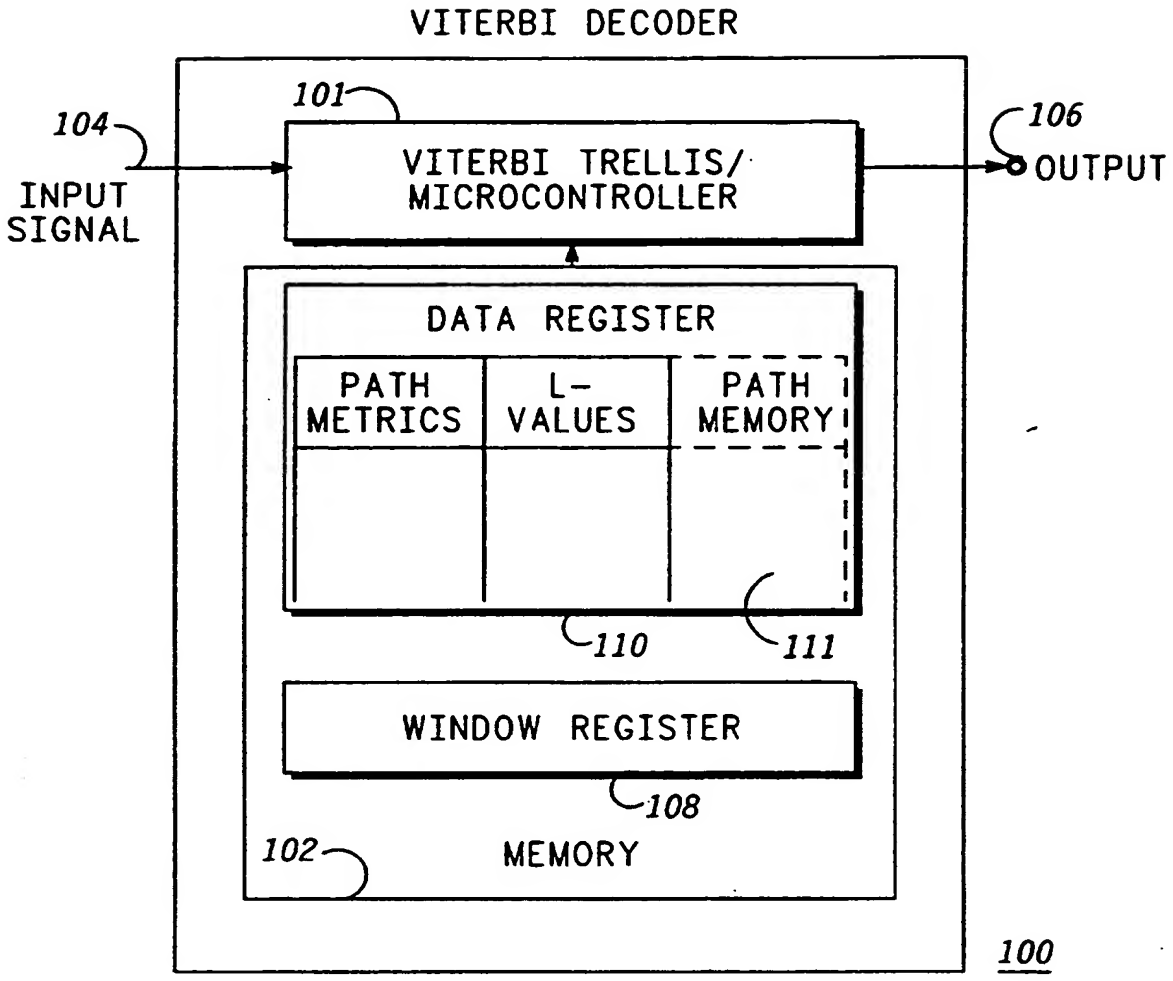


FIG. 5

112

PATH	L_{MIN} FOR ENTIRE PATH TO DATE
A	L_{MIN_A}
B	L_{MIN_B}
C	L_{MIN_C}

FIG. 6

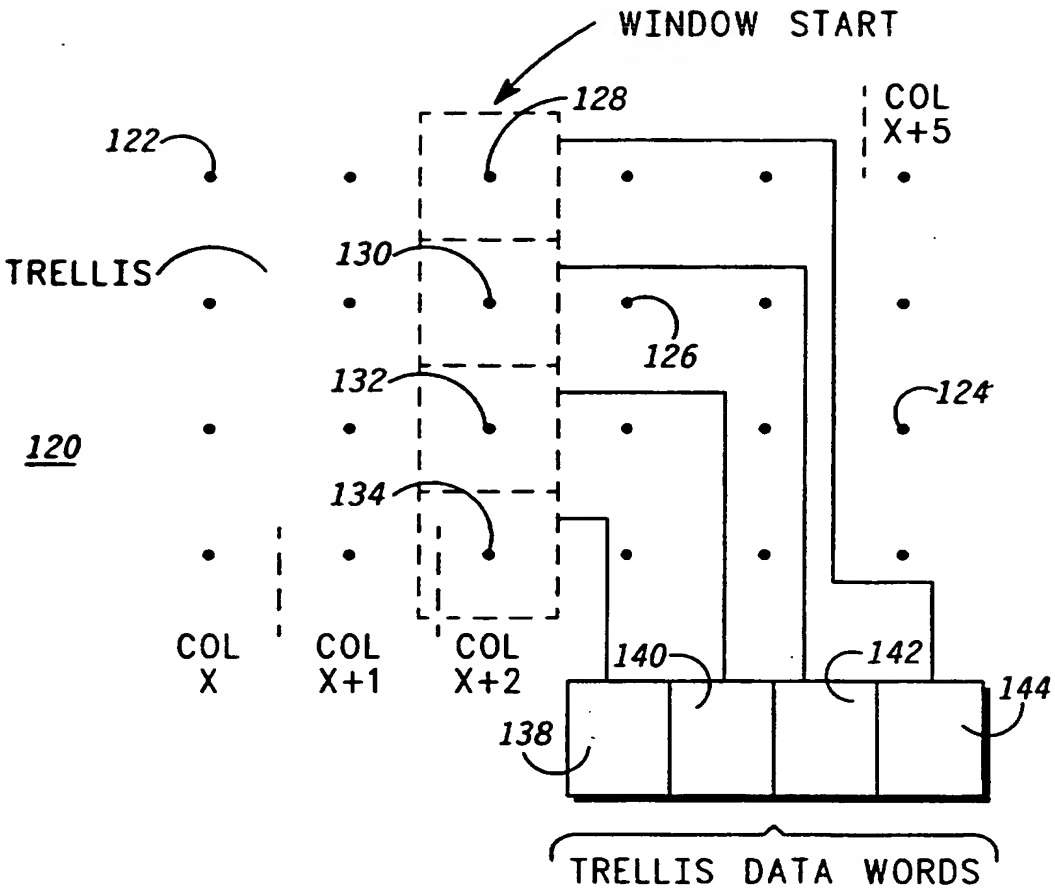


FIG. 7

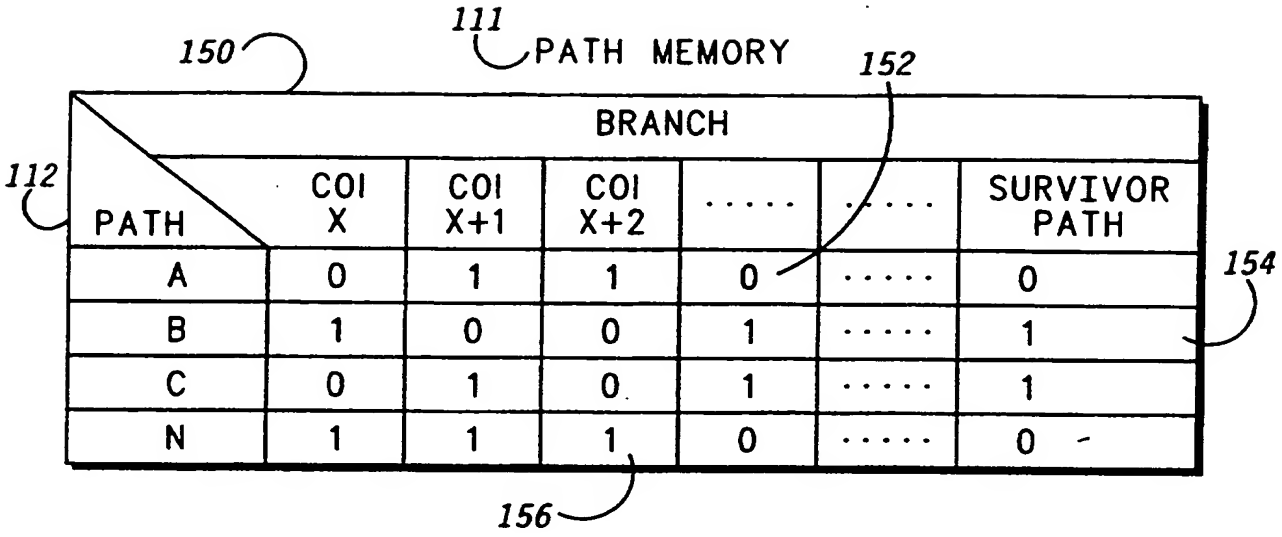


FIG. 8

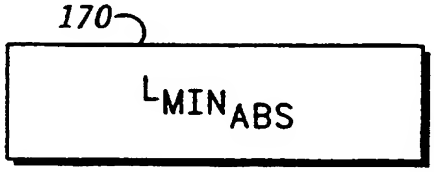


FIG. 9

2305827

VITERBI CO-PROCESSOR AND METHOD OF OPERATION THEREFOR

Field of the Invention

This invention relates, in general, to a Viterbi co-processor, and is particularly, but not exclusively, applicable to Viterbi co-processor operation, such as the implementation of calculations for minimum metric-distance likelihood searches required for Window Error Detection (WED).

Summary of the Prior Art

In modern communication systems, such as the pan-European Groupe Speciale Mobile (GSM) cellular communication system, forward error correction (FEC) of the physical radio channel is provided by means of signal encryption. More specifically, in the case of the GSM communication system, this encryption takes the form of convolutional coding.

The complex nature of these codes requires extensive computational (processor) power, with the optimum decoding process provided by a Viterbi decoding algorithm. In the Viterbi algorithm, decoded bits of information are output in a ratio of one output bit per input bit. However, the decoding process also provides additional information which can be utilised to allow multiple output bits (say N) to be generated per input bit, where each N-bits represents a confidence measure of the actual output bit. The use of this confidence measure is often referred to as "soft-decision" decoding. One particular method used to provide additional confidence information requires the output of metric-difference values (or "Likelihood" values, L) associated with each decision point along a "survivor path" of a Viterbi (state) trellis.

Viterbi co-processors already exist that can calculate required L-values for the purposes of soft-decision decoding, although some are limited in this respect to a "soft-output" mode (usually associated with Maximum

Likelihood Sequence Estimation (MSLE)) and are therefore not enabled for convolutional decoding, as required in multi-mode machines.

With respect to the Viterbi trellis (and algorithm), an output sequence is determined based upon the most likely path through a logical trellis of states consisting of logical "0"s and logical "1"s that are distributed over time. More particularly, at reception of a signal, a measurement of the most likely bit sequence for the signal is made by making distance measurements ("branch metric calculations") between received data bits and hypothesised data bits. In fact, the L-values provide a measurement of an absolute distance between two alternative branch metrics that converge to a decision point in the Viterbi state trellis along the survivor path. Furthermore, since paths through the trellis converge (with time) to convergence points, these convergence points present an opportunity to discard, at these junctures, all paths except the most likely. Specifically, the path offering the smallest distance (i.e. the largest branch metric) is the path retained, while the others are discarded. Subsequently, the lowest L-value (L_{MIN}) for the survivor path is used as an index to an array of threshold values, with a received speech (or data) frame marked bad and rejected if L_{MIN} is lower than a corresponding threshold.

As indicated above, the likelihood values can be used to provide "soft" output decisions for each bit input into the Viterbi algorithm. Indeed, an application for the Likelihood values is the so-called Soft Output Viterbi Algorithm (SOVA) that uses the L-values to iterate the survivor path decisions until convergence occurs on a final optimum path. Also, the L-values may be used to facilitate a reduced complexity (non-iterative) SOVA to provide a confidence measure of a whole (or portion) of the speech frame. An example of this form of reduced complexity SOVA arises in the Window Error Detection algorithm used in GSM half-rate coding.

In the GSM half-rate speech specification, there is a requirement to include the WED algorithm (associated with the Viterbi decoding algorithm used in decoding convolutional code applied to half-rate speech channels) which provides a mechanism for assessing quality of a decoded frame of speech. Particularly, the quality assessment (measurement) is

used to determine subsequent acceptance or rejection (erasure) of the decoded speech (or data) frame.

As will be appreciated, the most important bits in the half-rate speech frame are bits 73 to 97 (in combination, the so-called class 1a bits and the three (3) parity bits). These bits are important in the sense that they convey the most essential information about the speech samples contained in the transmitted information frame.

Furthermore, as will be appreciated, the WED function need only be applied to the most important bits to determine the reliability of the transmission of these bits. Indeed, the WED algorithm utilises L-values (produced as a by-product of the Viterbi branch metric calculation).

The problem with present Viterbi decoders is that they are unable to provide L_{MIN} values directly, and instead provide L-values that require further processing intervention (and hence greater data processing power and electrical power consumption) to calculate L_{MIN}. Additionally, the WED algorithm actually requires a moving average of L_{MIN} values to be stored in order to generate quality threshold information. Unfortunately, present (Viterbi) co-processor design is unable to cope with this additional requirement, so yet more post-Viterbi-stage processing (which clearly effects efficiency in implementing the WED function) must be performed off-line by a central processor. Also, existing co-processors perform L-value calculations for all elements of the Viterbi trellis, even though the GSM WED function, for example, only requires the function to be implemented on a certain part (bits 73 to 97 in the specific instance of the GSM WED function) of the input data frame. However, present-day techniques require the necessary storage of all L-values since prior to completing the decode of a complete frame, the survivor path through the trellis is unknown. Consequently, there is no way of knowing in advance which decision (convergence) points require the storage of L-values.

Clearly, the present methodology not only results in the performance of many unnecessary L-value calculations (for all bits output from the trellis) and increases the complexity of the Viterbi decoding (and WED function) process, but also requires significant control circuitry and substantial

memory storage facilities in the Viterbi decoder (typically implemented as a DSP on a semiconductor chip). As such, prior art methods are less efficient in terms of throughput (i.e. operations performed per second) and power consumption, and incur greater cost as a result of requiring the use of high frequency devices.

As such, a need exists for an improved method of Viterbi decoding that requires a lower volume of data processing.

Summary of the Invention

According to a first aspect of the present invention there is provided a method of calculating a metric-difference value of a survivor path for an encoded frame provided to a Viterbi decoder arranged to perform a decoding process on the encoded frame, the Viterbi decoder comprising: processor means for controlling the decoding process; a Viterbi state trellis responsive to the encoded frame and having a plurality of branches each having a plurality of logic states; and a memory, responsive to the processor means, for storing data generated during the decoding process and for storing an indication of a particular branch in the Viterbi trellis from which the metric-difference value is to be calculated; the method comprising the steps of: a) during a first pass of the decoding process, calculating path metrics for the encoded frame for a plurality of paths through the Viterbi state trellis, such that each of the plurality of paths comprises a logic state in each of said plurality of branches and such that an association between the logic state and a corresponding path metric is realised by the processor means and stored in the memory; b) in response to the indication and while calculating the path metrics, associating each logic state of the plurality of logic states in the particular branch with a path metric to produce a plurality of trellis data words, and storing these trellis data words in memory; c) at termination of the first pass, identifying the survivor path; d) determining a particular logic state in the particular branch through which the survivor path passed by using the association of at least some of the logic states of the survivor path; and e) executing a second pass of the decoding process for the encoded frame from the particular state in the particular branch by initially selecting and using a path metric from a trellis data word of the plurality of trellis data

words that corresponds to that particular state in order to calculate a minimum metric-difference value for the survivor path only.

In a preferred embodiment the minimum metric-difference value is stored in the memory and subsequently compared against newly calculated minimum metric-difference values for the survivor path in subsequent branches, such that only a numerically smallest value is retained in memory. Advantageously, this reduces the number of memory locations required to store calculated metric-difference values.

In a further embodiment, the step of determining is achieved by performing a trace-back of the survivor path using at least some of the associations stored in memory during the first pass, wherein a comparison of relative sizes between possible alternative associations for the survivor path in a branch determines the survivor path. Alternatively, the association indicates a relative directional displacement from a preceding logic state of the survivor path in a preceding branch of the Viterbi state trellis, and the step of determining is achieved by performing a trace-back of the survivor path using the relative directional displacement between logic states in at least some adjacent branches along the survivor path.

In another aspect of the present invention there is provided a Viterbi co-processor arranged to perform a decoding process on an encoded frame, the Viterbi co-processor comprising: processor means for controlling the decoding process; a Viterbi state trellis responsive to the encoded frame and having a plurality of branches each having a plurality of logic states; and a memory, responsive to the processor means, for storing data generated during the decoding process and for storing an indication of a particular branch in the Viterbi trellis from which the metric-difference value is to be calculated; the processor means further comprising: means for calculating path metrics for the encoded frame for a plurality of paths through the Viterbi state trellis during a first pass of the decoding process, wherein each of the plurality of paths comprises a logic state in each of said plurality of branches; means for generating and storing in the memory an association between the logic state and a corresponding path metric; means, response to the indication, for associating each logic state

of the plurality of logic states in the particular branch with a path metric to produce a plurality of trellis data words, and for storing these trellis data words in memory; means for identifying the survivor path at termination of the first pass; means, responsive to the association of at least some of the logic states of the survivor path, for determining a particular logic state in the particular branch through which the survivor path passed; means for executing a second pass of the decoding process for the encoded frame from the particular state in the particular branch; means for initially selecting and using a path metric from a trellis data word of the plurality of trellis data words that corresponds to that particular state; and means, responsive to the path metric selected from the trellis data word that corresponds to that particular state, for only calculating a minimum metric-difference value for the survivor path.

Exemplary embodiments of the present invention will now be described with reference to the accompanying drawings.

Brief Description of the Drawings

FIG. 1 is a representation of a prior art convolutional coder.

FIG. 2 shows a prior art Viterbi state trellis containing a multitude of possible paths.

FIG. 3 illustrates an association of likelihood (L-) values with logic states on the Viterbi state trellis of FIG. 2.

FIG. 4 is a prior art memory block used in association with the trellis of FIG. 2 and the mechanism of FIG. 3.

FIG. 5 is a block diagram of a Viterbi decoder according to the present invention.

FIG. 6 is a memory block used by the Viterbi decoder of FIG. 5 for storing L-values in accordance with a first preferred operating mechanism of present invention.

FIG. 7 illustrates tapping and storage of state values from a Viterbi state trellis in accordance with an alternate operating mechanism of the present invention.

FIG. 8 illustrates a memory block utilised by the Viterbi decoder of FIG. 5.

FIG. 9 shows a memory location for storage of an absolute L_{MIN} value for a Viterbi algorithm in accordance with the alternate operating mechanism of FIG. 7.

Detailed Description of a Preferred Embodiment

Referring to FIG. 1 there is shown a representation of a prior art convolutional coder 10 that is used to encode (prior to transmission) information bits of an input signal 12. More particularly, input bits are shifted on a bit-by-bit basis into register (memory) elements 14 and 16, at which point the register elements 14 and 16 are selectively tapped and logically combined (usually with and input bit). In the case of convolutional coding, logical combination typically takes the form of an exclusive-OR (XOR) function. With particular regard to FIG. 1, register element 16 is tapped and XORed in logic gate 18 with input bit 20. Also, register elements 16 is tapped and XORed in logic gate 22 with input bit 20 and the contents of register element 14. Logic gates 18 and 22 provide (at outputs 24 and 26, respectively) convolutionally encoded output data bits 28 and 30. The convolutional coder 10 of FIG. 1 is therefore said to have a constraint length of three (3), indicating the total number of bits of the input signal 12 used in the encoding operation, and a rate of 1/2, i.e. that every input bit produces two (2) output bits.

Although, for the sake of brevity and simplicity, only two register elements are illustrated, it will be appreciated that a convolutional coder will typically comprise several such register elements.

Now, FIG. 2 shows a prior art Viterbi state trellis 50 (containing a multitude of possible paths 52-56) that is arranged to decode a received convolutionally encoded signal 58. Each path 52-56 represents a possible logic transition between states with respect to time. Therefore, most trellis

points 59-66 in the Viterbi trellis 50 potentially provide a point of convergence. In the specific case of FIG. 2, the thickened line 57 represents a survivor path having an output ultimately selected at trellis point 64. Additionally, the illustrated Viterbi state trellis 50 shows a four-state code by way of example, although the numbers of states may be varied according to requirements, e.g. eight-state or sixteen-state codes may be utilised in some channel coding protocols. Points 64 and 66 represent possible survivor paths through the trellis and can therefore be loosely considered as outputs. Again, as will be understood, only a relatively few states and paths have been specifically identified for the sake of clarity, although many states and paths exist within the Viterbi trellis 50.

A mechanism for calculating likelihood (L-) values on the Viterbi state trellis of FIG. 2, is shown in FIG. 3. More particularly, logic states M and $M+2^{V-1}$ (hereinafter referred to by reference numerals 70 and 72, respectively) have calculated path (or survivor) metrics P_A and P_B associated respectively therewith and stored, respectively, in memory elements 76 and 78. In the particular instance of FIG. 3, logic states 70 and 72 converge to logic state 2M (hereinafter referred to by reference numeral 74) via branch (B) metrics 80 and 82 (or B_A and B_B), respectively. Therefore, logic state 74 also has a calculated path metric P_N associated therewith that is stored in memory element 84. Additionally, logic state 74 has a L-value (L_N) associated therewith and stored in memory element 86. Typically, the memory elements used to store the path metrics will be located in a memory block of the Viterbi decoder. As will be understood, the values of the path (survivor) metric and the L-value are calculated by the following equations:

$$P_N = \text{Maximum value of } \{(P_A + B_A), (P_B + B_B)\} \quad \text{(eqn. 1); and}$$

$$L_N = |(P_A + B_A) - (P_B + B_B)| \quad \text{(eqn. 2).}$$

FIG. 4 is a prior art memory block 90 used in association with the Viterbi state trellis of FIG. 2, and the mechanism for calculating path metrics and likelihood values of FIG. 3. Specifically, movement through the Viterbi trellis of FIG. 2 can also be considered as a transition between columns (or "branches") containing a plurality of logic states, with each logic state in each branch being located on a possible path through the Viterbi trellis.

As has been explained, since the final survivor path through the trellis is unknown until the completion of the decode of the complete frame, each L-value for each state in the trellis must also be stored (in addition to the retention in memory of the path metrics associated with each path). As such, the memory block 90 contains a vast array of storage elements 91-93 (typically indexed in some identifying manner, such as by way of state 95 and column/branch position 97).

Referring now to FIG. 5, there is shown a block diagram of a Viterbi decoder 100 according to the present invention. As will be appreciated, the Viterbi decoder 100 has signal processing capabilities and comprises a Viterbi trellis/microcontroller portion 101 and a memory portion 102 that is responsive to the Viterbi trellis/microcontroller portion 101 and which is used for data storage. The Viterbi trellis/microcontroller portion 101 (as typified by FIG. 2) is responsible for the operational control of the decoding process and data manipulation. More particularly, an encoded input signal 104 received by the Viterbi decoder is decoded by the Viterbi trellis/microcontroller portion 101 to produce a decoded output 106 indicative of survivor paths through the trellis. The memory portion 102 contains registers for storing information generated by the microcontroller portion 101 during the decoding process. More particularly, the memory portion 102 contains a data register 110 for storing and indexing data, e.g. recording path metrics, L-values and related information generated upon traversing the trellis. The memory portion 102 additionally includes a window register 108 used to define a window of N branches identified by a start-column (and optionally a finish-column). The window register 108 identifies a branch in the trellis at which to start metric-difference (L-) value calculations (e.g. start at "column x") and potentially a branch in the trellis at which to suspend/terminate such calculations (e.g. stop at "column x+5"). Furthermore, the N branches of the window represent the portion of the speech/data frame of the Viterbi state trellis that is subject to the WED function. In the absence of a finish-column, the metric-difference value calculations will commence at the start-column and continue throughout the subsequent branches (or columns) of the trellis. Optionally, the data register 110 may also comprise a path memory 111 for storing mapping bits (the function of which is specific to

an alternate embodiment of the present invention to be described subsequently).

In FIG. 6 there is shown, in greater detail, a portion of the data register 110 used (by the Viterbi decoder of FIG. 5) according to a first preferred operating mechanism of present invention. As well as containing an indexed array of calculated path metrics, the data register 110 stores only a minimum L-value (e.g. L_{MIN_A}) for each indexed path route 112 (e.g. path "A"). More particularly, the minimum L-value is the numerically smallest L-value for each of the survivor paths at the current branch in the trellis.

According to a first preferred operating mechanism of the present invention, L-values are stored in the L_{MIN} register 110 by the Viterbi trellis/microcontroller portion 101 through a "minimum wave method". Metric-difference (L-value) calculations commence (as defined by the start-column) at the start of the window, which start point may be at the beginning of the frame or at some later point. Then, for every branch in the window, the L-values are calculated for each decision point (and hence each possible path), and then compared against a L-value stored previously for each path. Only the lowest numeric L-value for each particular path (e.g. (L_{MIN_B}) is stored in the L_{MIN} register 110, which process may therefore involve the over-writing (substitution) of older but higher numeric L-values. As will be understood, it is possible for paths to converge at a convergence point (such as logic state 74) when traversing between branches. In this case, it is necessary to calculate two L-values for the two alternate converging paths, and then to compare the newly calculated L-values against previously stored L-values for the different paths previously taken to this convergence point. In this respect, it will be appreciated that in the present case only two path metrics (and hence two L-values) can potentially converge to a convergence point for the exemplary trellis arrangement illustrated.

In summary, when traversing between adjacent branches of a Viterbi state trellis, newly calculated L-values are compared with previously calculated L-values, with only the lowest numeric value retained as the L_{MIN} value for each path. Then, once the frame has been decoded and the

survivor path(s) identified, the L_{MIN} value stored for a desired survivor path can be immediately determined and read-out by the microcontroller portion 101. As a result of this method, a better quality numeric value for the metric-difference calculation is obtained through the selection of the absolute lowest L_{MIN} value over an entire path (or portion thereof). Indeed, use of a programmable register arrangement of the first preferred operating mechanism provides greater flexibility in decoding and allows adaptation of Viterbi decoders for future changes in coding algorithm, e.g. the so-called enhanced full-rate coding algorithms. Furthermore, as will be appreciated, by defining a window of interest in the Viterbi trellis, fewer calculation are required and a resultant power saving in a DSP therefore achieved.

In an alternate embodiment of the present invention, it has been realised that the survivor path through a Viterbi trellis can be identified independently of calculated L-values for the survivor path. Thus, WED in a Viterbi decoder is operated through a "two-pass method". The Viterbi algorithm is first executed (pass #1) with the calculation and storage of L-values suppressed/suspended.

FIG. 7 illustrates tapping and storage of state values from a Viterbi state trellis in accordance with the alternate operating mechanism of the present invention. Particularly, a portion 120 of the Viterbi state trellis has been illustrated, which portion 120 comprises a plurality of logic states 122-134 distributed amongst a plurality of branches (col x, col x+1, col x+2,...col x+5). A particular branch (in this case col x+2) has been identified as the start-column, with the particular logic states 128-134 contained therein tapped to provide respective trellis data words 138-144, comprising the path metrics for each state in branch x+2, directly associated with the start of the window. As will be appreciated, tapping of the logic states 128-134 in the Viterbi state trellis, in general, may be implemented by hard-wire taps or through a "state download" of logic state values from the branches, for example. Clearly, a state download offers greater flexibility with respect to window definition and is the preferred method. The data trellis words 138-144 (and their respective associations with the logic states in the branch of the start of the window) are stored in the data register 110.

As will be appreciated, all of the path metrics calculated during pass #1 of a frame through the Viterbi state trellis need not be stored, since some will be discarded or over-written and others may prove superfluous to the decoding algorithm. However, in a first implementation of the alternate embodiment of the present invention, it is necessary to record all path metrics from at least the start of the window to the point where a desired survivor path is determined. Alternatively, in a second implementation of the alternate embodiment of the present invention, the data register need only store a most recently calculated path metric for each path through the trellis together with an indication (stored in path memory 111) of which particular one of the two possible logic states in the preceding branch provided the route of the path. The nature of the path memory 111 will now be described in relation to FIG. 8.

Referring to FIG. 8, an exemplary structure of the path memory is illustrated. Paths 112 through the Viterbi state trellis are indexed against branches 150 (such as those identified as col x, col x+1, col x+2 and survivor path) of the trellis. Logical values 152-156, i.e. logical "0"s or logical "1"s (referred to as "pointers"), are associated with each path and branch in the path memory 111. More particularly, since (in the present case) a path passes through a state in a branch only derivable from one of two earlier logic states in a preceding branch, then a logical "0" or a logical "1" is attributed against a particular path and branch dependent upon a perceived direction from which the path emanates. For example, attribution of a logical "0" could indicate the movement from logic state "00" of the Viterbi state trellis 50 to logic state "01", rather than from logic state "10" to "01", whereas attribution of a logical "1" in the path memory would indicate the opposite. As such, attribution of pointers in the path memory 111 can also be considered as indicating the relative movement (directional displacement) of the path based on absolute values of the preceding logic states as compared against a present logic state for a state in a branch. Clearly, any appropriate form of logical attribution (pointer) indicating the relative nature of the higher or lower logic state in the preceding branch can be utilised in the path memory 111.

Storage of pointers as an indication of each survivor path is advantageous since only one bit of information need by stored to indicate the path direction, and only the most recent path metric for the path need be recorded in the data register. Consequently, there is a reduction in the amount of memory required to store the data generated by the decoding process.

On completion of pass #1, a desired survivor path is identified in accordance with usual, known practices that will be readily appreciated by the skilled addressee, e.g. the identification of the path having the smallest distance (largest branch metric) or, alternatively, the path that terminates at a given logical state, e.g. the logical "0" state in the Viterbi state trellis (as is the case for GSM channel decoding). At this point, the desired survivor path is "traced-back" to its corresponding state in the start-column (i.e. the first branch) of the identified window. Trace-back to the corresponding state in the first branch may be achieved through the realisation that the state having the smallest distance (largest path metric) of the two possible path metrics in an immediately preceding branch of the trellis represents the actual route of the desired survivor path (although this requires storage, during pass #1, of all path metrics for the trellis). Alternatively, trace-back may be achieved using the pointers stored in path memory 111 and the path metric of the survivor path (bearing in mind that the path metric of the survivor path is mathematically derived from previously calculated path metrics), i.e. trace-back is achieved by simply moving to (selecting) the appropriate logic state (of the two possible logical states) in the preceding branch in response to the direction indicated by the logical value of the pointer.

After trace-back, the trellis data words 138-144 recorded and associated with the first branch of the window during pass #1 are nominally re-loaded into the first branch of the window, and a portion of the Viterbi algorithm is executed a second time (pass #2) for the same frame (concentrating on the desired survivor path). During pass #2, metric-difference (L-value) calculations are performed for the desired survivor path only, and only a single absolute lowest $L_{MIN_{ABS}}$ value for the subsequent branches of the trellis is recorded and stored (which potentially results in a previous $L_{MIN_{ABS}}$ value being discarded and therefore

over-written). In fact, only the path metric associated with the desired survivor path and the corresponding state in the first branch of the identified window need be re-loaded.

As will be appreciated, during pass #2 and provided that all path metrics have been stored, there is no need to re-calculate path metrics for the desired survivor path because these have already been calculated during pass #1 and have been previously stored in memory, although a numeric comparison between the two possible path metric values would need to be made to determine the actual path. As such, the values of the path metrics subsequently encountered on the desired survivor path are retrieved directly. However, if the path memory 111 is utilised to store pointers and only the survivor path metric recorded, then path metric calculations will be required, which path metric calculations will necessarily commence at the first branch of the window identified by window register 108.

Selection between the first trace-back implementation (i.e. the storage of all path metrics) and the second implementation (i.e. the storage of pointers and only the most recent path metric) depends upon the operating characteristics and structure of the Viterbi decoder because there is a memory/power trade-off between these two alternative implementations (with the first requiring a larger memory capacity but potentially simpler processing).

FIG. 9 shows a memory location (or element) 170 for storage of an absolute $L_{MIN_{ABS}}$ value for a Viterbi algorithm in accordance with the alternate operating mechanism of FIG. 7, which memory location 170 may be located in the data register 110.

Implementation of the alternate embodiment provides substantial benefits and advantages with respect to window error detection and Viterbi decoding. Particularly, substantial memory storage space is saved in terms of both storage of path metric values and particularly L-values, since only one relatively small register is required to store the single $L_{MIN_{ABS}}$. Additionally, the alternate operating method substantially reduces the number of calculations performed by the DSP (Viterbi

decoder), and hence a substantial reduction in power consumption of the DSP results. Indeed, when employing the two-pass method for the WED function, there is an order of magnitude reduction in the number of calculation required, and an estimated power saving of ~70% for a GSM-type Viterbi half-rate code. As a direct consequence of the reduction in the number of calculation, the rate (and efficiency) of the Viterbi decoding process is increased substantially which therefore allows Viterbi decoders to be implemented on substantially lower frequency silicon devices (or the like). Furthermore, having alleviated the requirement for high frequency devices, a substantial cost saving and higher manufacturing productivity results.

Claims

1. A method of calculating a metric-difference value of a survivor path for an encoded frame provided to a Viterbi decoder arranged to perform a decoding process on the encoded frame, the Viterbi decoder comprising: processor means for controlling the decoding process; a Viterbi state trellis responsive to the encoded frame and having a plurality of branches each having a plurality of logic states; and a memory, responsive to the processor means, for storing data generated during the decoding process and for storing an indication of a particular branch in the Viterbi trellis from which the metric-difference value is to be calculated; the method comprising the steps of:

a) during a first pass of the decoding process, calculating path metrics for the encoded frame for a plurality of paths through the Viterbi state trellis, such that each of the plurality of paths comprises a logic state in each of said plurality of branches and such that an association between the logic state and a corresponding path metric is realised by the processor means and stored in the memory;

b) in response to the indication and while calculating the path metrics, associating each logic state of the plurality of logic states in the particular branch with a path metric to produce a plurality of trellis data words, and storing these trellis data words in memory;

c) at termination of the first pass, identifying the survivor path;

d) determining a particular logic state in the particular branch through which the survivor path passed by using the association of at least some of the logic states of the survivor path; and

e) executing a second pass of the decoding process for the encoded frame from the particular state in the particular branch by initially selecting and using a path metric from a trellis data word of the plurality of trellis data words that corresponds to that particular state in order to calculate a minimum metric-difference value for the survivor path only.

2. The method of calculating a metric-difference value according to claim 1, wherein the minimum metric-difference value is stored in the memory and subsequently compared against newly calculated minimum metric-difference values for the survivor path in subsequent branches, and wherein only a numerically smallest value is retained in memory.

3. The method of calculating a metric-difference value according to claim 1 or 2, wherein the step of determining is achieved by performing a trace-back of the survivor path using at least some of the associations stored in memory during the first pass, wherein a comparison of relative sizes between possible alternative associations for the survivor path in a branch determines the survivor path.

4. The method of calculating a metric-difference value according to claim 1 or 2, wherein the association indicates a relative directional displacement from a preceding logic state of the survivor path in a preceding branch of the Viterbi state trellis, and the step of determining is achieved by performing a trace-back of the survivor path using the relative directional displacement between logic states in at least some adjacent branches along the survivor path.

5. The method of calculating a metric-difference value according to any preceding claim, wherein the survivor path is identified as the path metric having the smallest distance.

6. The method of calculating a metric-difference value according to any one of claims 1 to 4, wherein the survivor path is identified as the path that terminates at a predetermined logical state in a predetermined branch of the Viterbi state trellis.

7. A Viterbi co-processor arranged to perform a decoding process on an encoded frame, the Viterbi co-processor comprising:

processor means for controlling the decoding process;

a Viterbi state trellis responsive to the encoded frame and having a plurality of branches each having a plurality of logic states; and

a memory, responsive to the processor means, for storing data generated during the decoding process and for storing an indication of a particular branch in the Viterbi trellis from which the metric-difference value is to be calculated;

the processor means further comprising:

means for calculating path metrics for the encoded frame for a plurality of paths through the Viterbi state trellis during a first pass of the

decoding process, wherein each of the plurality of paths comprises a logic state in each of said plurality of branches;

means for generating and storing in the memory an association between the logic state and a corresponding path metric;

means, response to the indication, for associating each logic state of the plurality of logic states in the particular branch with a path metric to produce a plurality of trellis data words, and for storing these trellis data words in memory;

means for identifying the survivor path at termination of the first pass;

means, responsive to the association of at least some of the logic states of the survivor path, for determining a particular logic state in the particular branch through which the survivor path passed;

means for executing a second pass of the decoding process for the encoded frame from the particular state in the particular branch;

means for initially selecting and using a path metric from a trellis data word of the plurality of trellis data words that corresponds to that particular state; and

means, responsive to the path metric selected from the trellis data word that corresponds to that particular state, for only calculating a minimum metric-difference value for the survivor path.

8. The Viterbi co-processor of claim 7, wherein the means for determining further comprises:

means for performing a trace-back of the survivor path using at least some of the associations stored in memory during the first pass; and

means for comparing relative sizes between possible alternative associations for the survivor path in a branch;

wherein the means for determining determines the survivor path in response to the relative sizes.

9. The Viterbi co-processor of claim 7, wherein the memory, in response to the processor means, stores a path comprising associations indicative of a relative directional displacement from a preceding logic state of the survivor path in a preceding branch of the Viterbi state trellis;

and wherein the means for determining the particular logic state in the particular branch through which the survivor path passed identifies

the particular logic state by performing a trace-back of the survivor path in response to the relative directional displacement between logic states in at least some adjacent branches along the survivor path.

10. A method of calculating and storing metric-difference values of at least one survivor path for an encoded frame provided to a Viterbi decoder arranged to perform a decoding process on the encoded frame substantially as hereinbefore described with reference to FIGs. 5 to 9 of the accompanying drawings.

11. A Viterbi co-processor for decoding an encoded frame substantially as hereinbefore described with reference to FIGs. 5 to 9 of the accompanying drawings.

Patents Act 1977 20 **Examiner's report to the Comptroller under Section 17 (The Search report)**	Application number GB 9519995.6
Relevant Technical Fields (i) UK Cl (Ed.N) H4P PRV (ii) Int Cl (Ed.6) H03M 13/00, H04L 1/00	Search Examiner MR B J SPEAR Date of completion of Search 16 NOVEMBER 1995
Databases (see below) (i) UK Patent Office collections of GB, EP, WO and US patent specifications. (ii) ONLINE: WPI, CLAIMS, JAPIO, USPATFULL, INSPEC	Documents considered relevant following a search in respect of Claims :- 1-11

Categories of documents

X: Document indicating lack of novelty or of inventive step.

Y: Document indicating lack of inventive step if combined with one or more other documents of the same category.

A: Document indicating technological background and/or state of the art.

P: Document published on or after the declared priority date but before the filing date of the present application.

E: Patent document published on or after, but with priority date earlier than, the filing date of the present application.

&: Member of the same patent family; corresponding document.

Category	Identity of document and relevant passages	Relevant to claim(s)
	NONE	

Databases: The UK Patent Office database comprises classified collections of GB, EP, WO and US patent specifications as outlined periodically in the Official Journal (Patents). The on-line databases considered for search are also listed periodically in the Official Journal (Patents).